UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,351	12/31/2001	Joseph P. Bratt	04860.P2686	7732

7590 09/09/2004

EXAMINER
O#BRIEN, BARRY J

ART UNIT	PAPER NUMBER
2183	4

James C. Scheller
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, CA 90025-1026

DATE MAILED: 09/09/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/038,351	BRATT ET AL.
	Examiner	Art Unit
	Barry J. O'Brien	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 December 2001 and 31 May 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-79* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-79* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-79 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Declaration as received on 5/31/2002.

Information Disclosure Statement

3. The Application Transmittal form received on 12/31/01 indicates that an IDS was to be filed with the application. However, no such IDS or accompanying citations was submitted.

Specification

4. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

5. The applicant is requested to review the specification and update the status of all co-pending applications made mention of, replacing attorney docket numbers with current U.S. application or patent numbers when appropriate.

Claim Objections

6. Claims 13, 15, 17, 19, 22, 42, 61-62, 67-68, 74-75 and 77 are objected to because of the following informalities:

a. A series of singular dependent claims is permissible in which a dependent claim refers to a preceding claim which, in turn, refers to another preceding claim. A claim that depends from a dependent claim should not be separated by any claim that does not also depend from said dependent claim. It should be kept in mind that a dependent claim may refer to any preceding independent claim. In general, applicant's sequence will not be changed. See MPEP § 608.01(n). See claims 36-43, where claim 42 improperly depends on claim 36, while claim 43 depends on claim 33. Also see where claims 61-62 improperly depend on claim 54, and claims 74-75 improperly depend on claim 58.

b. Claim 61 recites the limitation "configure according to an indicator" on its second line. Please correct the claim language to read "configured according to an indicator".

c. Claims 67 and 77 recite the limitation, "two complement" on their fourth lines. Please correct the claim language to read "two's complement" so as to be correct with the term in the art.

d. Claims 67 and 77 recite the limitation, "sign magnitude" on their third lines. Please correct the claim language to read "signed magnitude" so as to be correct with the term in the art.

e. Claim 68 recites the limitation "sign magnitude" on its fourth line. Please correct the claim language to read "signed magnitude" so as to be correct with the term in the art.

f. Claims 13, 15, 17, 19 and 22 are objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Here, the above claims all recite the limitation "A processing system comprising an execution unit". However, an execution unit is inherently a processing system, as it processes instructions by definition. Thus, the above claims to not further limit the subject matter of their parent claims.

Appropriate action is required.

Double Patenting

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

8. Claims 14, 16, 18, 23, 26, 27 and 33 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1, 4, 5, 1, 4, 5 and 11, respectively, of copending Application No. 10/038,478. Although the conflicting claims are not identical, they are not patentably distinct from each other because claims 14, 16,

18, 23, 26, 27 and 33 are obvious variations of claims 1, 4, 5 and 11 of the copending application.

9. For example, claims 14, 16 and 18 of the instant application comprise limitations within a microprocessor execution unit. Claims 1, 4 and 5, respectively, of the copending application contain the same limitations as the above claims in the instant application, but are comprised in a method for execution by a microprocessor. However, because the instant application has not claimed a specific execution unit, rather just "circuits" for performing the limitations, the microprocessor execution unit of the instant application is an obvious variation over the method for execution by a microprocessor, as some sort of "unit" is required to perform the method steps of the copending application.

10. Furthermore, claims 23, 26, 27 and 33 of the instant application comprise limitations within an execution unit in a microprocessor. Claims 1, 4, 5 and 11, respectively, of the copending application contain the same limitations as the above claims in the instant application, but are comprised in a method for execution by a microprocessor. However, because the instant application has not claimed a specific execution unit, rather just "means" for performing the limitations, the execution unit in a microprocessor of the instant application is an obvious variation over the method for execution by a microprocessor, as some sort of "means" are required to perform the method steps of the copending application.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 112

11. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

12. Claims 12, 20-22, 48-52, 56, 70-71 and 76-79 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

13. Claim 12 recites the limitation, "a sixth circuit coupled to the fifth circuit and the third circuit, in response to the microprocessor receiving the first instruction". It is unclear what is supposed to happen "in response to the microprocessor receiving the first instruction", as the limitation is associated with the "a sixth circuit" limitation, which does not make sense. Please correct the claim language to more clearly define the metes and bounds of the claim.

14. Claim 20 recites the limitation "the plurality of values" in its thirteenth line. There is insufficient antecedent basis for this limitation in the claim. Dependent claims 21-22 contain all the limitations of their parent claim, and thus are rejected for the same reasons as claim 20.

15. Claim 48 recites the limitation, "wherein further comprising" on its first line. It is unclear what is further comprised from the parent claim. Please correct the claim language to more clearly define the metes and bounds of the claimed invention. Dependent claims 49-52 contain all the limitations of their parent claim, and thus are rejected for the same reasons as claim 48

16. Claim 56 recites the limitation "the plurality of look-up units" in its first and second lines. There is insufficient antecedent basis for this limitation in the claim.

17. Claim 70 recites the limitation "the string" in its first line. There is insufficient antecedent basis for this limitation in the claim. For the purposes of this examination, the Examiner will assume "the string" refers to "the string of bits" of the parent claim of claim 70. Dependent claim 71 contains all the limitations of its parent claim, and thus is rejected for the same reasons as claim 70.

18. Claim 76 recites the limitation "the at least one data" in its fifth line. There is insufficient antecedent basis for this limitation in the claim. For the purposes of this examination, the Examiner will assume "the at least one data" refers to "the at least one escape data". Dependent claims 77-79 contain all the limitations of their parent claim, and thus are rejected for the same reasons as claim 76.

Claim Rejections - 35 USC § 102

19. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

20. Claims 1-79 are rejected under 35 U.S.C. 102(e) as being anticipated by Barry et al., U.S. Patent No. 6,397,324.

21. Regarding claim 1, Barry has taught an execution unit in a microprocessor, the execution unit comprising:

a. Look-up memory (431/433 of Fig.4),

b. A first circuit coupled to the look-up memory,

I. The first circuit, in response to the microprocessor receiving a first instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32), partitioning the look-up memory into a first plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67 and Col.11 lines 10-13),

II. The first circuit, in response to the microprocessor receiving a second instruction (see "L4TBL" on Col.11 lines 32-48), partitioning the look-up memory into a second plurality of look-up tables which are different from the first plurality of look-up tables (see Col.11 lines 33-36). Here, the look-up tables are different in size and number between the L2TBL and L4TBL instructions.

22. Regarding claim 2, Barry has taught an execution unit as in claim 1, wherein:

a. A total number of bits used by each entry in the first plurality of look-up tables is different from a total number of bits used by each entry in the second plurality of look-up tables (see Col.10 line 62 – Col.11 line 48). Here, the "size" field of both the L2TBL and the L4TBL instructions specifies whether the instruction is to use dual word, dual-half word, or dual byte data is to be used in the look table entries (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28). Thus a L2TBL and a L4TBL instruction can use different number of bits for each entry.

b. The microprocessor is a media processor formed in a monolithic semiconductor substrate, which comprises a memory controller (485 of Fig.4) for controlling DRAM memory (431/433 of Fig.4), said media processor being coupled to said

memory controller (see Fig.4). Here, while not taught explicitly, it is inherent that the processor of Barry is formed on a monolithic semiconductor substrate, as that is how digital circuits are created.

23. Regarding claim 3, Barry has taught an execution unit as in claim 1, wherein a total number of entries in each of the first plurality of look-up tables is different from a total number of entries in each of the second plurality of look-up tables (see Col.10 line 62 – Col.11 line 48). Here, Barry supports look-up table sizes of 256 entries for the L4TBL instruction (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries for the L2TBL instruction, each entry being an 8, 16 or 32-bit entry.

24. Regarding claim 4, Barry has taught an execution unit as in claim 1, wherein the look-up memory comprises a plurality of look-up units, and wherein the first circuit configuring the plurality of look-up units into a third plurality of look-up tables in response to the microprocessor receiving a third instruction (see "LTBL" on Col.10 lines 24-61). Here, the LTBL instruction partitions the look-up memory into a single look-up table (see Col.10 lines 24-61).

25. Regarding claim 5, Barry has taught an execution unit as in claim 4, wherein each of the third plurality of look-up units contains 256 8-bit entries (see Col.10 line 62 - Col.11 line 48). Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry.

26. Regarding claim 6, Barry has taught an execution unit as in claim 4, wherein a total number of entries in each of the third plurality of look-up tables is one of:

a. 256 (see Col.11 line 44),

b. 512,

c. 1024.

27. Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 6.

28. Regarding claim 7, Barry has taught an execution unit as in claim 4, wherein a total number of bits used by each entry in the third plurality of look-up tables is one of:

a. 8 ("two bytes" in Col.12 lines 14-28),

b. 16 ("two halfwords" in Col.12 lines 14-28),

c. 24.

29. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 7.

30. Regarding claim 8, Barry has taught an execution unit as in claim 1, further comprising:

a. A second circuit coupled to the look-up memory, the second circuit configured to receive a plurality of numbers (see An/Rz of Fig.6), in response to the microprocessor receiving the first instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32), the first plurality of look-up tables looking up simultaneously a plurality of entries, each of the plurality of entries being in one of the plurality of look-up tables and being pointed to by one of the plurality of numbers (see Col.9 lines 41-67 and Col.12 lines 14-27). Here, the L2TBL instruction, which is the

LTBL instruction modified to perform two look-up table look-ups (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.10 line 62 – Col.11 line 32). The above two pointers received point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

31. Regarding claim 9, Barry has taught an execution unit as in claim 1, further comprising:

a. A second circuit coupled to the look-up memory, the second circuit configured to receive a string of bits, in response to the microprocessor receiving the first instruction (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register address.

b. The second circuit generating a plurality of indices using a plurality of segments of bits in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two indices to entries into the look-up tables (see Col.10 line 62 – Col.11 line 32).

c. The first plurality of look-up tables (431/433 of Fig.4) looking up simultaneously a plurality of entries each of the plurality of entries being in one of the plurality of look-up tables and being pointed to by one of the plurality of indices (see Col.9 lines 41-67 and Col.12 lines 14-27). Here, the above two pointers received point

to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

32. Regarding claim 10, Barry has taught an execution unit as in claim 9, further comprising:

a. A third circuit coupled to the look-up memory, the third circuit combining the plurality of entries into a first result (see Col.11 lines 10-32). Here, the two outputted entries from each look-up table are combined and stored in register Rt, with each entry stored in one half of register Rt.

33. Regarding claim 11, Barry has taught an execution unit as in claim 10, further comprising:

a. A forth circuit coupled to the second circuit, the forth circuit configured to receive a plurality of data elements specifying the plurality of segments in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register addresses.

34. Regarding claim 12, Barry has taught an execution unit as in claim 10, further comprising:

a. A fifth circuit coupled to the second circuit, the fifth circuit configured to receive at least one format (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up

table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

b. A sixth circuit coupled to the fifth circuit and the third circuit, in response to the microprocessor receiving the first instruction (see Col.11 lines 10-32),

c. The fifth circuit formatting the string bits into a least one escape data using the at least one format format (see Col.11 lines 10-32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction.

d. The sixth circuit combing the at least one escape data with the first result into a second result (see Col.10 line 62 – Col.11 line 32). Here, the two outputted entries from each look-up table are sign-extended (i.e. combined with the escape data) if necessary, and then combined and stored in register Rt, with each entry stored in one half of register Rt.

35. Regarding claim 13, Barry has taught a processing system comprising an execution unit as in claim 1 (see Fig.4).

36. Regarding claim 14, Barry has taught a microprocessor execution unit (see Fig.4) comprising:

a. A plurality of look-up tables (431/433 of Fig.4),

b. A first circuit configured to accept a first plurality of numbers (see An/Ri of Fig.4 or An/Rz of Fig.8), each of the first plurality of numbers pointing to one of a plurality of entries, each of the plurality of entries being in one of the plurality of look-up tables (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

c. A second circuit configured to accept a second plurality of numbers (see Rs of Fig.4 or Rte/Rto of Fig.8),

d. A third circuit coupled to the first circuit, the second circuit, and the plurality of look-up tables, the third circuit, in response to the microprocessor receiving a single instruction, replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers (see Col.9 lines 41-62 and Col.12 lines 14-27).

37. Regarding claim 15, Barry has taught a processing system comprising an execution unit as in claim 14 (see Fig.4).

38. Regarding claim 16, Barry has taught an execution unit in a microprocessor (see Fig.4), the execution unit comprising:

a. A plurality of look-up tables (431/433 of Fig.4),

b. A first circuit coupled to the plurality of look-up tables and a Direct Memory Access (DMA) controller, the first circuit, in response to the microprocessor receiving a single instruction, replacing at least one entry in at least one of the plurality of look-up tables with at least one data element using the DMA controller (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20). The entries are updated via the memory interface unit (485 of Fig.4), which is a DMA controller (see Col.7 lines 50-62).

39. Regarding claim 17, Barry has taught a processing system comprising an execution unit as in claim 16 (see Fig.4).

40. Regarding claim 18, Barry has taught an execution unit in a microprocessor (see Fig.4), the execution unit comprising:

a. A plurality of look-up tables (431/433 of Fig.4),

b. A first circuit coupled to the plurality of look-up tables and a Direct Memory Access (DMA) controller, the first circuit, in response to the microprocessor receiving a single instruction, replacing at least one entry for each of the plurality of look-up tables with a plurality of data elements using the DMA controller (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20). The entries are updated via the memory interface unit (485 of Fig.4), which is a DMA controller (see Col.7 lines 50-62).

41. Regarding claim 19, Barry has taught a processing system comprising an execution unit as in claim 18 (see Fig.4).

42. Regarding claim 20, Barry has taught an execution unit in a microprocessor comprising:

a. A plurality of look-up tables (431/433 of Fig.4),

b. A first circuit coupled to the plurality of look-up tables, the first circuit configured to receive a string of bits (see Col.10 line 62 – Col.11 line 32),

c. A second circuit coupled to the plurality of look-up tables and the first circuit, the second circuit configured to receive a plurality of data elements, in response to the

microprocessor receiving a single instruction, the second circuit generating a plurality of indices using the plurality of data elements and the string of bits, the plurality of look-up tables looking up simultaneously a plurality of entries using the plurality of indices (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The above two indices point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file (see Col.9 lines 41-67, Col.11 lines 1-13 and Col.12 lines 14-27).

d. A third circuit coupled to the plurality of look-up tables, the third circuit combining the plurality of values into a first result (see Col.11 lines 10-32). Here, the two outputted entries from each look-up table are combined and stored in register Rt, with each entry stored in one half of register Rt.

43. Regarding claim 21, Barry has taught an execution unit as in claim 20, further comprising:

a. A fifth circuit coupled to the second circuit, the fifth circuit configured to receive at least one format (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up

table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

b. A sixth circuit coupled to the fifth circuit and the third circuit, in response to the microprocessor receiving the single instruction, the fifth circuit formatting the string of bits into at least one escape data using the at least one format, and the sixth circuit combining the at least one escape data with the first result into a second result (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction. Then, the two outputted entries from each look-up table are sign-extended (i.e. combined with the escape data) if necessary, and then combined and stored in register Rt, with each entry stored in one half of register Rt.

44. Regarding claim 22, Barry has taught a processing system comprising an execution unit as in claim 21 (see Fig.4).

45. Regarding claim 23, Barry has taught an execution unit in a microprocessor (see Fig.4), the execution unit comprising:

a. Means for receiving a first plurality of numbers (see An/Ri of Fig.4 or An/Rz of Fig.8) and a second plurality of numbers (see Rs of Fig.4 or Rte/Rto of Fig.8), each of the first plurality of numbers pointing to one of a plurality of entries, each

of the plurality of entries being in one of a plurality of look-up tables (431/433 of Fig.4) (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

b. Means for replacing simultaneously the plurality of entries in the plurality of look-up tables with the second plurality of numbers (see Col.9 lines 41-62 and Col.12 lines 14-27),

c. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "S2TBL" on Col.12 lines 13-27).

46. Regarding claim 24, Barry has taught an execution unit as in claim 23, wherein the first plurality of numbers are received form a first entry in a register file (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27), and the second plurality of numbers are received form a second entry in the register file (see Col.9 line 25 – Col.10 line 20 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction specifies, using even/odd addressing, two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). Thus, the execution unit "receives"

the first plurality of numbers from a first entry in a register file (An + Rz). Furthermore, the S2TBL instruction specifies two pieces of data denoted by odd and even addresses (each data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

47. Regarding claim 25, Barry has taught an execution unit as in claim 24, wherein the single instruction specifies indices of the first (An and Rz of Fig.8, see also Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27) and second entries (Rt in Fig.8, see also Col.9 line 25 – Col.10 line 20 and Col.11 line 64 – Col.12 line 27) in the register file.

48. Regarding claim 26, Barry has taught an execution unit in a microprocessor (see Fig.4), the execution unit comprising:

a. Means for replacing at least one entry in at least one of a plurality of look-up units (431/433 of Fig.4) in a microprocessor unit with at least one number using a Direct Memory Access (DMA) controller (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20). The

entries are updated via the memory interface unit (485 of Fig.4), which is a DMA controller (see Col.7 lines 50-62).

b. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "S2TBL" on Col.12 lines 13-27).

49. Regarding claim 27, Barry has taught an execution unit in a microprocessor (see Fig.4), the execution unit comprising:

a. Means for replacing at least one entry for each of a plurality of look-up units in a microprocessor with a plurality of numbers using a Direct Memory Access (DMA) controller (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). The S2TBL instruction further specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20). The entries are updated via the memory interface unit (485 of Fig.4), which is a DMA controller (see Col.7 lines 50-62).

b. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "S2TBL" on Col.12 lines 13-27).

50. Regarding claim 28, Barry has taught an execution unit as in claim 27, wherein a single index encoded in the instruction specifies a location of the at least one entry in the plurality of look-up units (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction, which is the STBL instruction modified to perform two look-up table stores (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.12 lines 14-27). Thus, each instruction contains an encoded pointer that is an index into a corresponding look-up table.

51. Regarding claim 29, Barry has taught an execution unit as in claim 27, wherein a single index encoded in the instruction specifies a total number of the at least one entry in the plurality of look-up units (see Col.10 line 62 - Col.11 line 48). Here, Barry supports three instructions (STBL, S2TBL and S4TBL) that each specify a different number of entries in the look-up tables based on the instruction itself.

52. Regarding claim 30, Barry has taught an execution unit as in claim 27, wherein a source address of the plurality of numbers is specified in an entry of a register file (see Col.12 lines 14-28). The S2TBL instruction specifies two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20). Thus because the index of the register file is specified in the instruction, and the source address of the plurality of numbers is the index of that entry in the register file, the source address of the plurality of numbers is specified in an entry in the register file.

53. Regarding claim 31, Barry has taught an execution nit as in claim 30, wherein the single instruction specifies an index of the entry in the register file (see Col.12 lines 14-28). The S2TBL instruction specifies the register indices (see Fig.8A) of two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

54. Regarding claim 32, Barry has taught an execution unit as in claim 27, wherein an identity number encoded in the single instruction specifies the DMA controller (see Col.9 lines 25-52 and Col.11 line 64 – Col.12 line 27). Here, the S2TBL instruction specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers to an entry in one of the look-up tables (see Col.12 lines 14-27). Because the pointers are "encoded" in the instruction, and because all accesses to the look-up tables use the DMA controller (see Col.7 lines 50-62), the instruction inherently specifies the DMA controller.

55. Regarding claim 33, Barry has taught an execution unit in a microprocessor, the execution unit comprising:

a. Means for receiving a plurality of numbers (see An/Rz of Fig.6). Here, the L2TBL instruction, which is the LTBL instruction modified to perform two look-up table look-ups (see Col.9 lines 53-62), specifies two base registers (An.H1 and An.H0) and two offsets (Rze and Rzo) to create two pointers (a plurality of "numbers") to an entry in one of the look-up tables (see Col.10 line 62 – Col.11 line 32).

b. Means for partitioning look-up memory into a plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67),

c. Means for looking up simultaneously a plurality of elements from the plurality of look-up tables (431/433 of Fig.4), each of the plurality of elements being in one of the plurality of look-up tables and being pointed to by one of the plurality of numbers (see Col.9 lines 41-67 and Col.12 lines 14-27). Here, the above two pointers received point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

d. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32).

56. Regarding claim 34, Barry has taught an execution unit as in claim 33, wherein the means for receiving a plurality of numbers comprises:

a. Means for partitioning a string of bits into a plurality of segments to generate the plurality of numbers (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register address.

57. Regarding claim 35, Barry has taught an execution unit as in claim 34, wherein the single instruction specifies format information in which the plurality of numbers are stored in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of

bits within the Rz register (see Fig.6A), and formats them into two segments of data, Rze and Rzo based on the even and odd source register address (see Col.10 line 62 – Col.11 line 32).

58. Regarding claim 36, Barry has taught an execution unit as in claim 33, wherein the look-up memory comprises a plurality of look-up units (431/433 of Fig.4), and wherein the means for partitioning look-up memory comprises:

a. Means for configuring the plurality of look-up units into the plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67).

59. Regarding claim 37, Barry has taught an execution unit as in claim 33, wherein the string of bits is received from an entry of a register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register (see Fig.6A), and formats them into two segments of data, Rze and Rzo based on the even and odd source register address (see Col.10 line 62 – Col.11 line 32).

60. Regarding claim 38, Barry has taught an execution unit as in claim 37, wherein the single instruction specifies an index of the entry (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register (see Fig.6A), and formats them into two segments of data, Rze and Rzo based on the even and odd source register address (see Col.10 line 62 – Col.11 line 32).

61. Regarding claim 39, Barry has taught an execution unit as in claim 33, further comprising:

a. Means for storing the plurality of elements in an entry of a register file (see Col.10 line 62 – Col.11 line 32).

62. Regarding claim 40, Barry has taught an execution unit as in claim 39, wherein the single instruction specifies an index of the entry (see Rt of Fig.6A and Col.10 line 62 – Col.11 line 32).

63. Regarding claim 41, Barry has taught an execution unit as in claim 39, wherein the single instruction specifies format information in which the plurality of elements are stored in the entry (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies that the plurality of data entries are stored at even and odd addresses into the register file, thus specifying the format.

64. Regarding claim 42, Barry has taught an execution unit as in claim 36, wherein each of the plurality of look-up units comprises 256 8-bit entries (see Col.10 line 62 - Col.11 line 48). Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry.

65. Regarding claim 43, Barry has taught an execution unit as in claim 33, wherein the single instruction specifies a total number of entries contained in each of the plurality of look-up tables (see Col.10 line 62 - Col.11 line 48). Here, Barry supports three instructions (STBL, S2TBL and S4TBL) that each specify a different number of entries in the look-up tables, as well as the size of each entry.

66. Regarding claim 44, Barry has taught an execution unit as in claim 44, wherein the total number of entries is one of:

a. 256 (see Col.11 line 44),

b. 512,

c. 1024.

67. Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry. Because the claim has been

written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 44.

68. Regarding claim 45, Barry has taught an execution unit as in claim 33, wherein the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables (see Col.12 lines 14-28). Here, the "size" field of the S2TBL instruction (see Fig.8A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

69. Regarding claim 46, Barry has taught an execution unit as in claim 45, wherein the total number of bits is one of:

a. 8 ("two bytes" in Col.12 lines 14-28),

b. 16 ("two halfwords" in Col.12 lines 14-28),

c. 24.

70. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 46.

71. Regarding claim 47, Barry has taught an execution unit in a microprocessor, the execution unit comprising:

a. Means for receiving a string of bits (see Col.10 line 62 – Col.11 line 32),

b. Means for generating a plurality of indices using a plurality of segments of bits in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into

two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables.

c. Means for look up simultaneously a plurality of entries from a plurality of look-up tables (431/433 of Fig.4) using the plurality of indices (see Col.9 lines 41-67, Col.11 lines 1-13 and Col.12 lines 14-27). Here, the above two indices point to elements in the look-up tables, and the L2TBL instruction subsequently reads the elements pointed to and stores them in a corresponding entry in the register file.

d. Means for combining the plurality of entries into a first result (see Col.11 lines 10-32). Here, the two outputted entries from each look-up table are combined and stored in register Rt, with each entry stored in one half of register Rt.

e. Wherein the above means operate in response to the microprocessor receiving a single instruction (see "L2TBL" on Col.10 line 62 – Col.11 line 32).

72. Regarding claim 48, Barry has taught an execution unit as in claim 47, wherein further comprising:

a. Means for receiving a plurality of data elements specifying the plurality of segments in the string of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register addresses.

73. Regarding claim 49, Barry has taught an execution unit as in claim 48, wherein the plurality of data elements are received from an entry in a register file (see Col.10 line 62 –

Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register addresses.

74. Regarding claim 50, Barry has taught an execution unit as in claim 49, wherein the single instruction specifies an index of the entry in the register file (see Col.12 lines 14-28). The L2TBL instruction specifies the register indices (see Fig.6A) of two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.11 lines 10-32).

75. Regarding claim 51, Barry has taught an execution unit as in claim 48, further comprising:

a. Means for receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo based on the even and odd source register address. Thus, the L2TBL instruction can be considered a bit pointer.

76. Regarding claim 52, Barry has taught an execution unit as in claim 51, further comprising:

a. Means for generating a new bit pointer using the first result (see Col.10 line 62 – Col.11 line 32). Here, because the L2TBL instruction both reads and writes to the register file, when the first result is written into the register file, it is read on

subsequent instructions, thus creating a new bit pointer for those instructions to use.

77. Regarding claim 53, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving an offset, wherein the plurality of indices are determined using the offset and the plurality of segments of bits (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables.

78. Regarding claim 54, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for partitioning look-up memory into the plurality of look-up tables before said looking-up (see Col.7 lines 54-62 and Col.9 lines 63-67 and Col.11 lines 10-13).

79. Regarding claim 55, Barry has taught an execution unit as in claim 54, wherein the look-up memory comprises a plurality of look-up units, and wherein the means for partitioning look-up memory comprises:

a. Means for configuring the plurality of look-up units into the plurality of look-up tables (see Col.7 lines 54-62 and Col.9 lines 63-67 and Col.11 lines 10-13).

80. Regarding claim 56, Barry has taught an execution unit as in claim 69, wherein each of the plurality of look-up units comprises 256 8-bit entries (see Col.10 line 62 - Col.11 line 48).

Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry.

81. Regarding claim 57, Barry has taught an execution unit as in claim 47, wherein the single instruction specifies a total number of entries contained in each of the plurality of look-up tables (see Col.10 line 62 - Col.11 line 48). Here, Barry supports three instructions (STBL, S2TBL and S4TBL) that each specify a different number of entries in the look-up tables, as well as the size of each entry.

82. Regarding claim 58, Barry has taught an execution unit as in claim 57, wherein the total number of entries is one of:

a. 256 (see Col.11 line 44),

b. 512,

c. 1024.

83. Here, Barry supports look-up table sizes of 256 (see Col.11 line 44) and 64000 (see Col.11 lines 13-14) entries, each entry being an 8, 16 or 32-bit entry. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 58.

84. Regarding claim 59, Barry has taught an execution unit as in claim 47, wherein the single instruction specifies a total number of bits used by each entry contained in the plurality of look-up tables (see Col.12 lines 14-28). Here, the "size" field of the S2TBL instruction (see Fig.8A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

85. Regarding claim 60, Barry has taught an execution unit as in claim 59, wherein the total number of bits is one of:

a. 8 ("two bytes" in Col.12 lines 14-28),

b. 16 ("two halfwords" in Col.12 lines 14-28),

c. 24.

86. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 60.

87. Regarding claim 61, Barry has taught an execution unit as in claim 54, wherein the plurality of look-up tables are configured according to an indicator in an entry in a register file (see Col.11 lines 11-32). Here, the data type and the instruction type define how the look-up tables will be configured, and because the data type is determined from the data in the register file, it can be considered an indicator in the register file.

88. Regarding claim 62, Barry has taught an execution unit as in claim 61, wherein the single instruction specifies an index of the entry in the register file (see Col.12 lines 14-28). The S2TBL instruction specifies the register indices (see Fig.8A) of two pieces of data denoted by even and odd addresses (each piece of data is considered a number) stored in the register file at Rte and Rto that will be written into the look-up table entries pointed to by the pointers created (see Col.10 lines 5-20).

89. Regarding claim 63, Barry has taught an execution unit as in claim 47, wherein the means for combining the plurality of entries comprises:

a. Means for selecting valid data from the plurality of entries (see Col.11 lines 10-27). Here, "double-word" data is not supported by the L2TBL instruction, and thus is not selected for combining and subsequent storing into the register file.

90. Regarding claim 64, Barry has taught an execution unit as in claim 63, further comprising:

a. Means for generating an indicator indicating whether none of the plurality of entries is valid (see Col.11 lines 5-9). Here, the results being indeterminate is considered to indicate that none of the entries were of a valid data type.

91. Regarding claim 65, Barry has taught an execution unit as in claim 63, wherein the valid data is selected according to priorities of the look-up tables from which the plurality of entries are looked up (see Col.11 lines 10-32). Here, the "priorities" of the look-up tables are considered to be those entries that are valid have priority over non-valid, indeterminate entries (see Col.11 liens 5-9). Thus, valid entries are determined using "priorities" of the look-up tables.

92. Regarding claim 66, Barry has taught an execution unit as in claim 63, wherein the means for combining the plurality of entries further comprises:

a. Means for formatting the valid data according to a type of the valid data (see Col.11 lines 10-27). Here, the data is further formatted by sign extension if it is of types "dual half-word" or "dual-byte", and not sign extended if its of the other valid type ("dual word").

93. Regarding claim 67, Barry has taught an execution unit as in claim 66, wherein the type of the valid data is one of:

a. Zero fill,

b. Sign magnitude (see Col.11 lines 20-27),

c. Two complement.

94. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 67.

95. Regarding claim 68, Barry has taught an execution unit as in claim 67, further comprising:

a. Means for retrieving a sign bit from the string of bits for the valid data, wherein the first result is obtained by formatting the valid data using the sign bit when the type of the valid data is sign magnitude (see Col.11 lines 20-27). Here, when data of a valid type is to be sign extended, it is formatted to the correct size by sign-extending the sign bit.

96. Regarding claim 69, Barry has taught an execution unit as in claim 47, wherein an entry in the plurality of entries contains:

a. Information indicating whether the entry is valid (see Col.11 lines 10-27). Here, "double-word" data is not supported by the L2TBL instruction, and thus an unsupported data type in an entry of the look-up table inherently indicates that it is not valid, and a supported data type indicates that it is valid.

b. Information indicating a type of the entry (see Col.11 lines 10-27). Again, data of a supported type in an entry of the look-up table inherently is an indication of the type of the data in the entry.

c. Information indicating a number of bits of a code word to be decoded (see Col.11 lines 10-27). Here, the type of data defines the length of the data that is to be read out of the look-up table.

97. Regarding claim 70, Barry has taught an execution unit as in claim 47, wherein the string is received from an entry in a register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables.

98. Regarding claim 71, Barry has taught an execution unit as in claim 70, wherein the single instruction specifies an index of the entry in the register file (see Col.10 line 62 – Col.11 line 32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo.

99. Regarding claim 72, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving a first number indicating a position of a last bit of input in the string of bit (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28). Thus, the size of the data indicates where the last bit of the input string will be in relation to the register size.

100. Regarding claim 73, Barry has taught an execution unit as in claim 72, further comprising:

a. Means for generating an indicator indicating whether any bit after the last bit of input is used in obtaining the first result (see Col.11 lines 10-32). Here, there is inherently an indicator which indicates if the data needs sign-extension, which would indicate that bits after the last bit (i.e. the sign-extension bits) have been used in creating the first result.

101. Regarding claim 74, Barry has taught an execution unit as in claim 58, further comprising:

a. Means for generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code (see Col.15 lines 13-17 and Col.15 line 63 – Col.16 line 4).

102. Regarding claim 75, Barry has taught an execution unit as in claim 74, wherein the predetermined code represents an end of block condition (see Col.15 lines 13-17 and Col.15 line 63 – Col.16 line 4).

103. Regarding claim 76, Barry has taught an execution unit as in claim 47, further comprising:

a. Means for receiving at least one format (see Col.12 lines 14-28). Here, the "size" field of the L2TBL instruction (see Fig.6A and Col.12 lines 66-67) specifies whether the instruction is to use dual word, word, dual-half word or dual byte data is used in the look-up table entries, and thus specifies a format (see Col.10 line 62-Col.11 lines 27 and Col.12 lines 14-28).

b. Means for formatting the string of bits into at least one escape data according to the at least one format (see Col.11 lines 10-32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction.

c. Means for combining the at least one data and the first result into a second result. (see Col.10 line 62 – Col.11 line 32). Here, the two outputted entries from each look-up table are sign-extended (i.e. combined with the escape data) if necessary, and then combined and stored in register Rt, with each entry stored in one half of register Rt.

104. Regarding claim 77, Barry has taught an execution unit as in claim 76, wherein one of the at least one format is for data of a type which is one of:

a. Zero fill,

b. Sign magnitude (see Col.11 lines 20-27),

c. Two complement.

105. Because the claim has been written in the alternative format, only one of the alternative limitations is required to be taught by the prior art, and thus Barry has taught the limitations of claim 77.

106. Regarding claim 78, Barry has taught an execution unit as in claim 76, wherein the at least one format is received from an entry of a register file (see Col.10 line 62-Col.11 lines 27

and Col.12 lines 14-28). Here, the format is specifies in the "size" field of the instruction, and then the actual format (data of a certain data type) is read out from an entry in the register file.

107. Regarding claim 79, Barry has taught an execution unit as in claim 78, wherein the single instruction specifies an index of the entry in the register file (see Col.11 lines 10-32). Here, the L2TBL instruction specifies a string of bits within the Rz register, and partitions them into two segments of data, Rze and Rzo, which represent index offsets. These offsets are added to two base registers (An.H1 and An.H0) to form two indices into two look-up tables. The output for the look-up tables at these indices is formatted, i.e. sign-extended, if necessary according to the "size" field of the instruction.

Conclusion

108. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

109. Ansari, U.S. Patent No. 6,553,486, has taught load and store vector instructions for reading from and storing to a look-up table.

110. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Barry J. O'Brien whose telephone number is (703) 305-5864. After October 12th, 2004, the examiner can be reached at (571) 272-4171. The examiner can

normally be reached on Mon.-Fri. 6:30am-4:00pm, with the exception of first Friday of every bi-week.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached at (703) 305-9712, or at (571) 272-4162 on or after October 12th, 2004. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

111. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Barry J. O'Brien
Examiner
Art Unit 2183

BJO
9/7/2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100